Установка

Локальная установка.

Open Server

OpenServer является портативным программным комплексом и не требует установки. Сборку можно разместить на внешнем жёстком диске или флэш-накопителе, это позволит использовать OpenServer на любом компьютере, который отвечает системным требованиям.

https://ospanel.io/

Дистрибутив OpenServer представляет собой самораспаковывающийся архив в формате 7ZIP (расширение .exe).

Запустите исполняемый файл дистрибутива и выберите путь для распаковки файлов.

Скопируйте папку проекта в OpenServer/domains.

Запуск приложения.

Для запуска приложения используйте файл OpenServer.exe. При наличии возможности программу следует запускать только от имени администратора. После старта программы вы увидите красный флажок в трее Windows (область возле системных часов). Чтобы включить непосредственно сам веб-сервер нажмите на флажок, далее выберите пункт меню [Меню → Запустить].

Если сервер не запускается, перейдите к разделу Решение проблем данного справочного руководства.

Теперь проект доступен в браузере по адресу fam/.

Установка на сервере.

Для загрузки данный проекта на сервер использовался FTP-клиент FileZilla.

Вам нужно обладать хостингом, логином и паролем для подключения по FTP, хостинговая компания должна дать Вам эту информацию.

В этом случае мы заполняем панель быстрого подключения.

Поле Порт: можно оставить незаполненным, если вы не используете специфический порт для подключения.

Если подключение прошло успешно, скопируйте проект в директорию вашего сайта.

Необходимый минимум системных ресурсов:

* Хостинг с поддержкой MySQL, PHP 5.6
* Объём необходимой памяти 53 МБ.
* Импорт базы данных

Конечный этап установки локальной/на сервере

Зайдите в интерфейс PHPMyAdmin (при локальной установке находится в меню OpenServer) и создайте базу данных. После создания, выберите созданную базу данных и нажмите кнопку «Import», которая размещена в верхнем меню.

Перед Вами появится окно, в котором нужно нажать на кнопку «Выберите файл» и выбрать экспортированных файл с базой данных сайта. После чего нажать кнопку «Go».

После импорта базы данных на экране должно появиться сообщение, что импорт прошел успешно «Import has been successfully finished». На этом импорт базы данных закончен.

Развёртывание среды разработки

Выполните локальную установку приложения.

Node.js

Node или Node.js — программная платформа, основанная на движке [V8](https://ru.wikipedia.org/wiki/V8_(%D0%B4%D0%B2%D0%B8%D0%B6%D0%BE%D0%BA_JavaScript)) (транслирующем [JavaScript](https://ru.wikipedia.org/wiki/JavaScript" \o "JavaScript) в машинный код), превращающая JavaScript из узкоспециализированного языка в язык общего назначения.

https://nodejs.org/

Установите Node.js

Откройте консоль операционной системы и перейдите в папку с проектом, установленную в OpenServer.

Выполните команду npm install, для установки необходимых компонентов.

Выполните команду gulp watch, для запуска сборщика пакетов.

Рабочая среда развёрнута, теперь Вы можете менять исходный код, представленный в папке public, все изменения будут автоматически обрабатываться сборщиком.

После окончания разработки остановите работу сборщика в консоли и выполните команду gulp --production.

Выполнение этих действий соберёт финальную версию приложения.

Техническая часть

Клиентская часть приложения (Front end)

Приложение основано на реактивном фреймворке Vue.js.

Vue (произносится /vjuː/, примерно как view) — это прогрессивный фреймворк для создания пользовательских интерфейсов. В отличие от фреймворков-монолитов, Vue создан пригодным для постепенного внедрения. Его ядро в первую очередь решает задачи уровня представления (view), что упрощает интеграцию с другими библиотеками и существующими проектами. С другой стороны, Vue полностью подходит и для создания сложных одностраничных приложений (SPA, Single-Page Applications), если использовать его совместно с [современными инструментами](https://ru.vuejs.org/v2/guide/single-file-components.html) и [дополнительными библиотеками](https://github.com/vuejs/awesome-vue#libraries--plugins).

https://ru.vuejs.org/

Система состоит из отдельных подключаемых модулей.

Для создания каркаса страниц и модулей, вместо HTML, использовался Jade, из-за его упрощённого синтаксиса, а для написания таблиц стилей вместо CSS, использовался SASS.

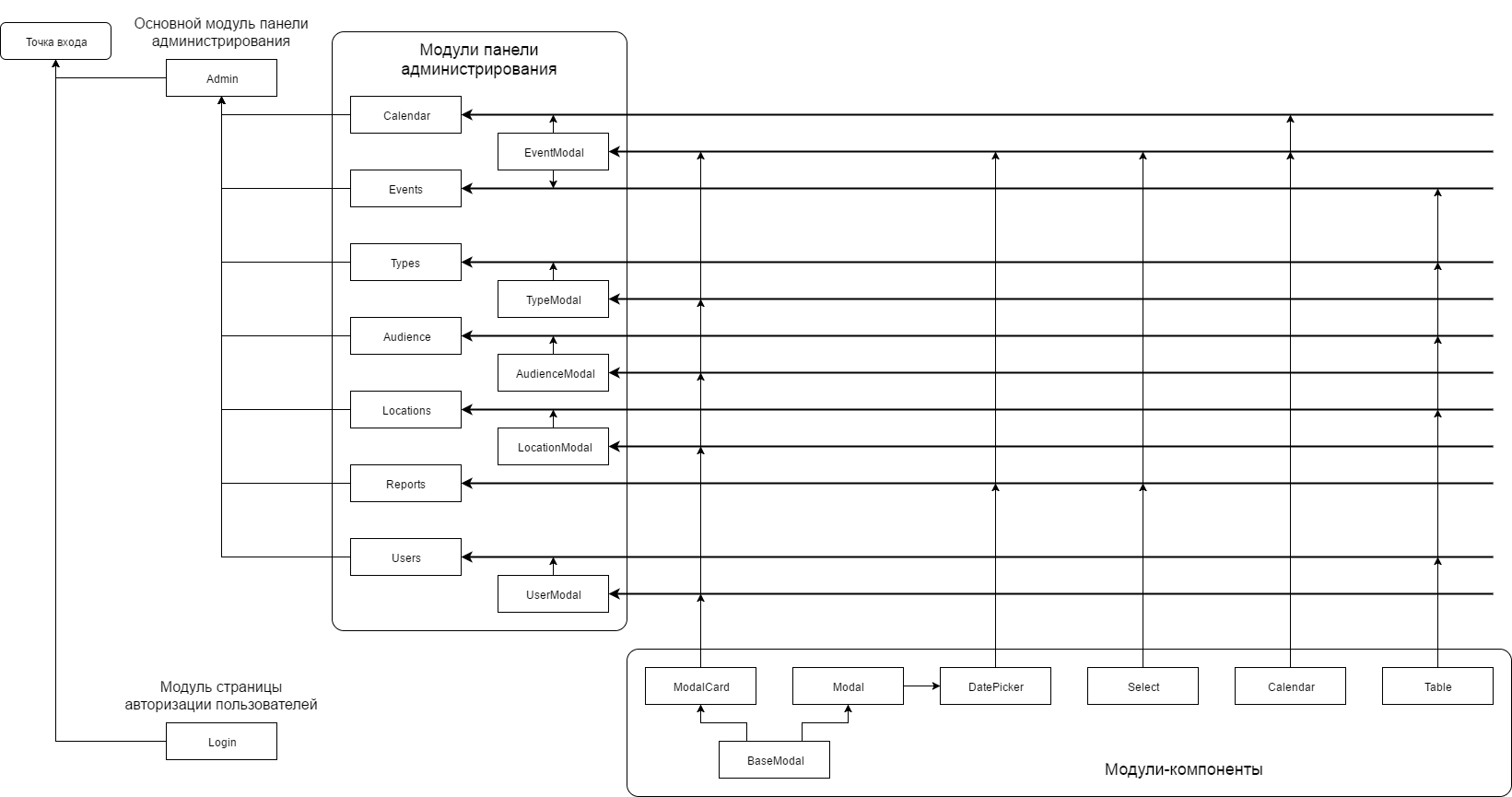
Разработка логики клиентской части системы велась с использованием стандарта ES-2015, языка [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) .

Для сборки модулей в [JavaScript](https://ru.wikipedia.org/wiki/JavaScript) , поддерживаемый браузерами, используется сборщик Laravel-Elixir и Webpack.

Для компиляции Jade и SASS в привычные для браузера форматы, в систему сборки были установлены соответствующие компиляторы.

Обзор модулей

Все модули содержатся в папке public/src.



Модули-компоненты (находятся в public/src/components)

Модули-компоненты описывают базовые элементы приложения, которые часто используются в большинстве крупных модулей.

Table.vue

Таблица.

Компонент загружает данные из таблицы в базе данных и показывает их пользователю в удобочитаемой форме. Доступен поиск, сортировка, выбор размера таблицы, переключение страниц и направление запроса на удаление записи в базе данных. Поиск работает одновременно по всем полям таблицы. При поиске, сортировке, выборе размера и удалении записи – номер страницы сбрасывается.

BaseModal.vue

Базовый модуль модального окна.

Все модули модальных окон расширяют этот компонент.

Modal.vue

Модальное окно.

Компонент выводить переданный в него контент в модальном окне.

Расширяет модуль BaseModal

ModalCard.vue

Модальное окно – карточка.

Компонент выводить переданный в него контент в модальном окне. В модальном окне присутствуют кнопки “Отмена” и “Сохранить”, которые передают запросы через систему слушателей, для управления содержимом модального окна.

Расширяет модуль BaseModal.

Select.vue

Компонент для множественного или единичного выбора значений из списка.

Поддерживает поиск и загрузку данных с сервера. Написан с использование библиотеки selectize.js, которая в свою очередь написана на jQuery.

Calendar.vue

Календар мероприятий.

Выводит список всех мероприятий в виде графика, в котором ось абсцисс является временной шкалой, а ось ординат – список всех мест проведения мероприятий. Доступно перемещение по временной шкале при помощи колёсика мыши и методом перетаскивания (drag-and-drop). Мероприятия выделяются жёлтым цветом, если ожидаемое количество посетителей превышает допустимое количество в месте проведения, привязанном к данному мероприятию. Красным цветом выделяются мероприятия, которые проходят в одном месте, в одно время.

DatePicker.vue

Компонент для выбора даты и времени.

Предоставляет удобный интерфейс для выбора даты и времени. В компоненте присутствуют отдельные вкладки для выбора месяца, дня, времени, минут и часов.

Включает в себя модуль Modal.

Панель администрирования

Точкой входа для системы панели администрирования является файл admin.js (находится в public/src/).

Здесь создаётся экземпляр Vue и VueRouter, необходимый для навигации между страницами-модулями. А также подключается главный модуль панели Admin.vue.

Модули панели администрирования (находятся в public/src/admin)

EventModal.vue

Модальное окно для редактирования данных мероприятия.

Включает в себя модули ModalCard, Calendar, Select и DatePicker.

Предоставляет возможность редактирования и создания мероприятий. При выборе мероприятия в календаре, который прикреплён к форме, данные мероприятия загрузятся с сервера в форму.

TypesModal.vue

Модальное окно для редактирования данных типа мероприятия.

Включает в себя модули ModalCard.

Предоставляет возможность редактирования и создания типов мероприятий. После добавления типа, к нему можно привязать уже существующие или новые мероприятия.

AudienceModal.vue

Модальное окно для редактирования данных типа аудитории.

Включает в себя модули ModalCard.

Предоставляет возможность редактирования и создания типов аудитории. После добавления типа, к нему можно привязать уже существующие или новые мероприятия.

LocationModal.vue

Модальное окно для редактирования данных места проведения мероприятия.

Включает в себя модули ModalCard.

Предоставляет возможность редактирования и создания мест проведения мероприятий. После добавления места, к нему можно привязать уже существующие или новые мероприятия.

UserModal.vue

Модальное окно для редактирования данных пользователя.

Включает в себя модули ModalCard.

Предоставляет возможность редактирования и создания учётных записей пользователей. После добавления пользователя, можно войти в систему используя новые данные для авторизации.

Admin.vue

Основной модуль панели администрирования.

Содержит меню и основное рабочее пространство.

Index.vue

Страница с информацией о системе.

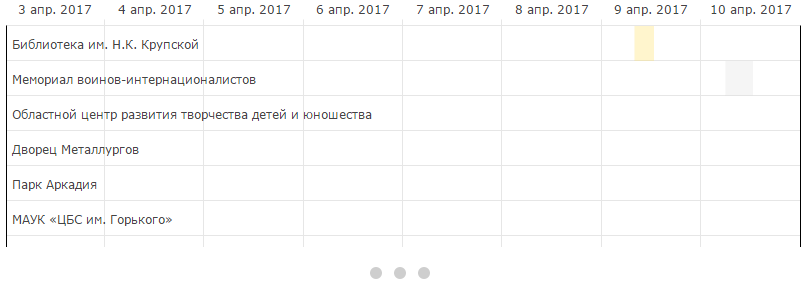
Calendar.vue

Страница с календарём мероприятий.

Включает в себя модули Calendar и EventModal.

При выборе мероприятия в календаре, откроется форма для редактирования записи.

При сохранении данных в форме, календарь автоматически обновиться.



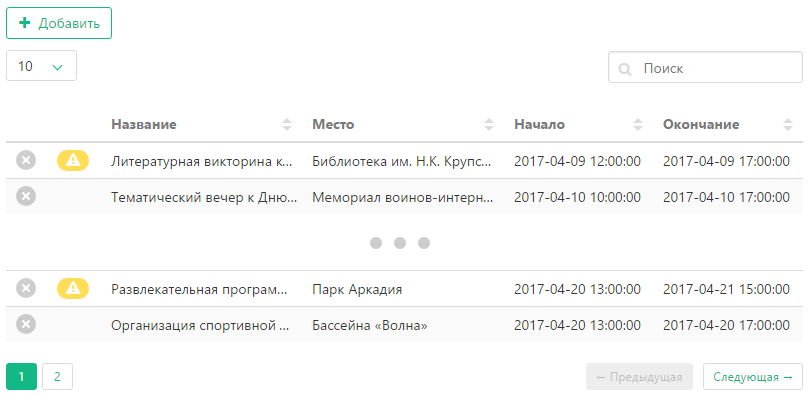
Events.vue

Страница со списком мероприятий.

Включает в себя модули Table и EventModal.

При выборе записи в таблице, откроется форма редактирования данных мероприятия. Мероприятия, на которых ожидаемое количество человек, превышает максимально допустимое количество в месте проведения, отмечаются жёлтым знаком предупреждения. Красным знаком отмечаются мероприятия, которые проходят в одном месте, в одно время. Конфликт мероприятий более приоритетен превышению допустимого количества посетителей, поэтому красный знак предупреждения всегда перекрывает жёлтый.

При сохранении данных в форме, таблица с событиями автоматически обновится.



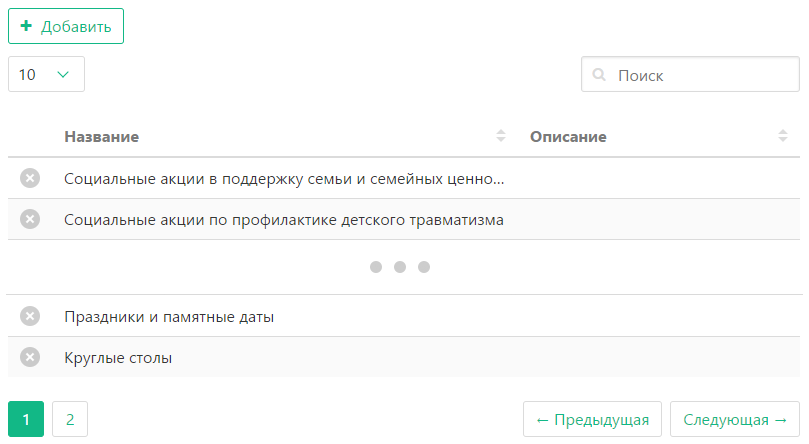
Types.vue

Страница со списком всех типов мероприятий.

Включает в себя модули Table и TypeModal.

При выборе записи в таблице, откроется форма редактирования данных типа мероприятия.

При сохранении данных в форме, таблица с типами мероприятий автоматически обновится.



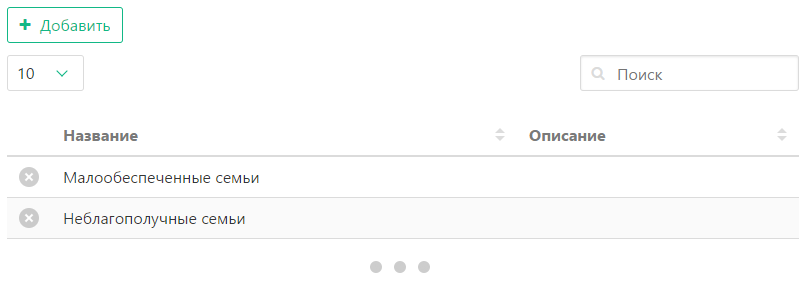
Audience.vue

Страница со списком всех типов аудитории.

Включает в себя модули Table и AudienceModal.

При выборе записи в таблице, откроется форма редактирования данных.

При сохранении данных в форме, таблица с типами автоматически обновится.



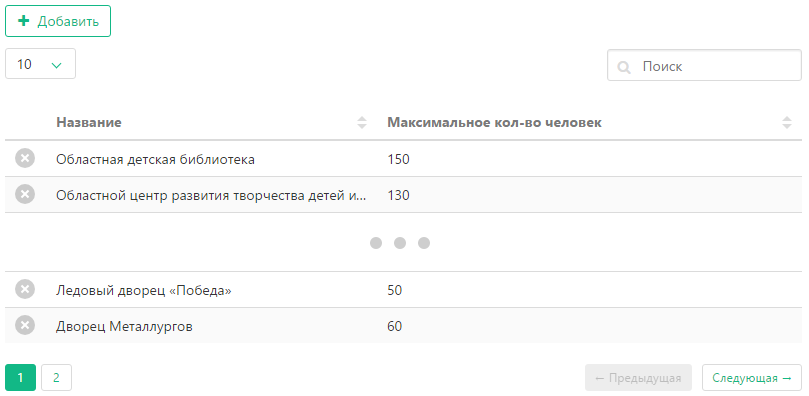
Locations.vue

Страница со списком всех мест проведения мероприятий.

Включает в себя модули Table и LocationModal.

При выборе записи в таблице, откроется форма редактирования данных места проведения мероприятия.

При сохранении данных в форме, таблица с местами проведения мероприятий автоматически обновится.



Reports.vue

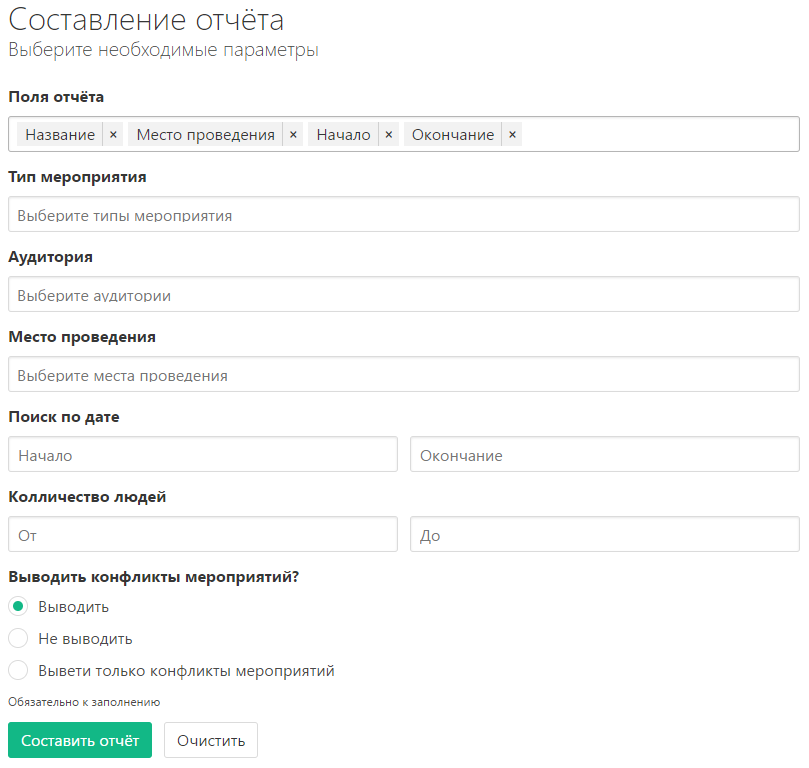
Страница для составления отчетов.

Включает в себя модули Select и DatePicker.

На странице предоставлен удобный набор опций для составления отчёта. Здесь можно указать:

* Список полей, в том порядке, в котором они войдут в финальную версию отчёта.
* Типы мероприятий.
* Типы аудитории.
* Места проведения мероприятий.
* Временной отрезок, за который необходимо составить отчёт. Указание обоих точек временного диапазона необязательно, если указанно только окончание диапазона, будут выбраны все мероприятия до этой даты, и наоборот, если выбрано только начало диапазона – будут выбраны все мероприятия, начиная с этой даты.
* Диапазон ожидаемого количества посетителей.
* Способ вывода конфликтов мероприятий.

Сформированный отчёт будет открыт на новой вкладке, где его можно будет распечатать.



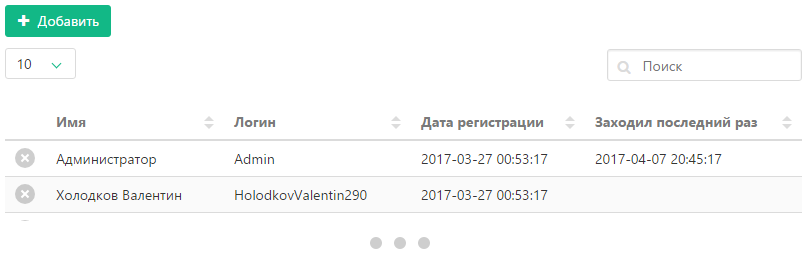
Users.vue

Страница со списком всех пользователей.

Включает в себя модули Table и UserModal.

При выборе записи в таблице, откроется форма редактирования данных пользователя. В форме можно указать новый пароль пользователя, старый пароль не выводиться в форме в целях безопасности.

При сохранении данных в форме, таблица с данными пользователей автоматически обновиться.



Страница входа

Точкой входа для страницы входа является файл login.js (находится в public/src/).

Здесь создаётся экземпляр Vue, также подключается главный модуль страницы Login.vue.

Модули страницы входа (находятся в public/src/login)

Login.vue

Страница для авторизации пользователей.

Серверная часть приложения (Back end)

Общение клиентской и серверной части приложения основывается на системе ajax запросов.

Порядок выполнения запроса для получения данных с сервера.

* Из клиентской части направляется запрос к серверу.

Url запроса состоит из названия контроллера и названия метода. Так же в url могут присутствовать дополнительные параметры, которые будут переданы в метод.

* На сервере создаётся экземпляр обработчика ошибок, который в случае ошибки формирует ответ с необходимыми HTTP заголовками и возвращает его
* Сервер создаёт экземпляр обработчика ссылок, и извлекает из url название контроллера и метода.
* Сервер проверяет наличие контроллера и метода, в случае ошибки возвращает ответ HTTP с кодом 500.
* Сервер создаёт экземпляр объекта запроса, в который помещает данные переданные пользователем.
* Сервер запускает метод контроллера, передавая ему экземпляр объекта запроса.
* Метод, выполнив необходимые действия на сервере, возвращает ответ.
* Сервер формирует из данных, которые вернул метод, JSON строку и создаёт экземпляр объекта ответа.
* Сервер помещает JSON строку в экземпляр объекта ответа, расставляет необходимые HTTP заголовки и возвращает его.
* Клиентская часть принимает ответ от сервера и расшифровывает полученные данные.

Описание таблиц базы данных

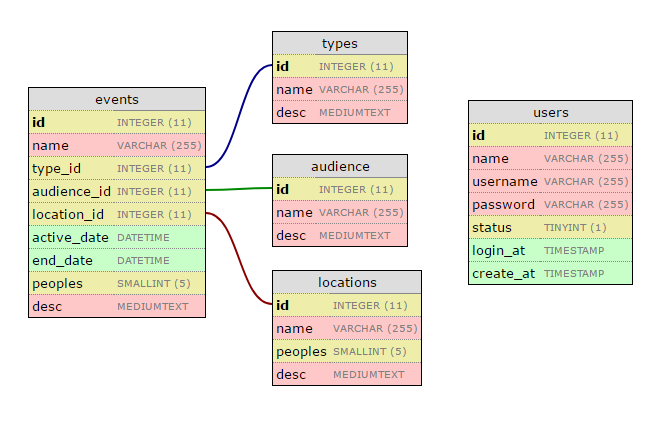


Таблица events

Таблица с данными событий.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Размер | Назначение | Примечание |
| id | INTEGER | 11 | Внешний ключ | Первичный ключ |
| name | VARCHAR | 255 | Название события |  |
| type\_id | INTEGER | 11 | Тип события | Внешний ключ к таблице types |
| audience\_id | INTEGER | 11 | Аудитория | Внешний ключ к таблице audience |
| location\_id | INTEGER | 11 | Id места проведения | Внешний ключ к таблице locations |
| active\_date | DATETIME |  | Начало активности |  |
| end\_date | DATETIME |  | Окончание активности |  |
| peoples | SMALLINT | 5 | Количество людей |  |
| desc | TEXT |  | Описание |  |

Таблица types

Таблица с типами событий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Размер | Назначение | Примечание |
| id | INTEGER | 11 | Внешний ключ | Первичный ключ |
| name | VARCHAR | 255 | Название типа |  |
| desc | TEXT |  | Описание |  |

Таблица audience

Таблица с типами аудитории

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Размер | Назначение | Примечание |
| id | INTEGER | 11 | Внешний ключ | Первичный ключ |
| name | VARCHAR | 255 | Название типа |  |
| desc | TEXT |  | Описание |  |

Таблица locations

Таблица с данными мест проведения мероприятий

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Размер | Назначение | Примечание |
| id | INTEGER | 11 | Внешний ключ | Первичный ключ |
| name | VARCHAR | 255 | Название места |  |
| peoples | SMALLINT | 5 | Количество людей |  |
| desc | TEXT |  | Описание |  |

Таблица users

Таблица с данными пользователей

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Имя | Тип | Размер | Назначение | Примечание |
| id | INTEGER | 11 | Внешний ключ | Первичный ключ |
| name | VARCHAR | 255 | Имя |  |
| username | VARCHAR | 255 | Логин |  |
| password | VARCHAR | 255 | Пароль |  |
| status | TINYINT | 1 | Статус |  |
| login\_at | TIMESTAMP |  | Время авторизации |  |
| create\_at | TIMESTAMP |  | Время создания записи |  |